Privacy-Preserving and Feature Importance-based Incentive Mechanism in Vertical Federated Learning

Sindhuja Madabushi (presenter), Haider Ali, Ahmad Khan, Mengmeng Gu, and Jin-Hee Cho

Department of Computer Science, Virginia Tech, Blacksburg VA

ACM CAPWIC 2024
6 April 2024

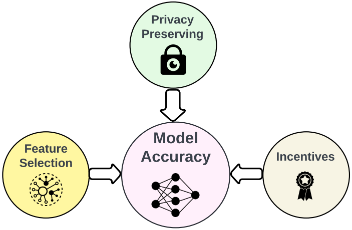
Outline

- **Motivation & Goal**
- **Related Work**
- **Problem Statement & Contributions**
- **Background: Horizontal vs. Vertical Federated Learning**
- **System Model**
- **Proposed Framework**
  - Privacy-Preserving Mechanism: Differential Privacy
  - Importance-based Feature Selection
  - Proposed Incentive Mechanism
- **Experimental Setup**
- **Preliminary Results**
  - Accuracy without Differential Privacy (DP)
  - Impact of Differential Privacy
  - Comparison with Existing Schemes
- **Key Findings & Future Work**

Motivation & Goal

Why incentive mechanisms (IMs) for VFL?
Clients may withdraw from the federation due to the following challenges:

- **Privacy concerns**
- **Spurious features**
- **Resource constraints**

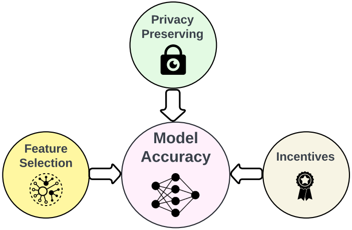


Motivation & Goal

Why incentive mechanisms (IMs) for VFL?

Clients may withdraw from the federation due to the following challenges:

- **Privacy concerns**
- **Spurious features**
- **Resource constraints**



Why don't the existing IM solutions for VFL work?

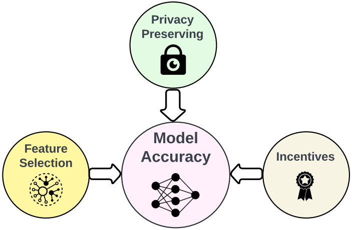
No IM for VFL has considered both privacy-preserving and feature importance-based learning in their IM solutions.

Motivation & Goal

Why incentive mechanisms (IMs) for VFL?

Clients may withdraw from the federation due to the following challenges:

- **Privacy concerns**
- **Spurious features**
- **Resource constraints**



Why don't the existing IM solutions for VFL work?

No IM for VFL has considered both privacy-preserving and feature importance-based learning in their IM solutions.

Goal: Develop an attack-resistant, robust vertical federated learning via incentive mechanisms that consider privacy-preserving and feature importance by achieving:

- high prediction accuracy
- a required level of privacy-preserving
- high efficiency under resource-constrained clients

Related Work

- **Privacy-Preserving Feature Selection (FS) in VFL**
  - Additive secret-sharing for FS (Zhang et al., 2022)
  - Stochastic dual-gate for the probability of features (Li et al., 2023)
  - Communication-efficient FS in VFL (Castigia et al., 2023)
  - IM based on bankruptcy problem (Khan et al., 2023)
- **Incentive Mechanisms (IMs) in VFL**
  - Feature importance-based IM (Tan et al., 2023)
  - Economic mechanism between clients (Yang et al., 2023)
  - Truthful IM (Lu et al., 2023)
  - Fairness-aware IM (Shi et al., 2022)
  - Reputation-based IM using Shapley value (Thi et al., 2021).

Related Work

- **Privacy-Preserving Feature Selection (FS) in VFL**
  - Additive secret-sharing for FS (Zhang et al., 2022)
  - Stochastic dual-gate for the probability of features (Li et al., 2023)
  - Communication-efficient FS in VFL (Castigia et al., 2023)
  - IM based on bankruptcy problem (Khan et al., 2023)
- **Incentive Mechanisms (IMs) in VFL**
  - Feature importance-based IM (Tan et al., 2023)
  - Economic mechanism between clients (Yang et al., 2023)
  - Truthful IM (Lu et al., 2023)
  - Fairness-aware IM (Shi et al., 2022)
  - Reputation-based IM using Shapley value (Thi et al., 2021).
- **Limitations**
  - Lack of studies considering *both* feature selection *and* privacy-preserving for incentive mechanism.
  - Insufficient incentive mechanism research for VFL.

Problem Statement & Contributions

We aim to develop a lightweight incentive mechanism that rewards clients who contribute to increasing prediction accuracy based on important features and preserving privacy. The reward function is given by:

$$\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$$

where $\mathcal{T}_i$ is the reward for client i, $\mathcal{I}$ is the performance contribution and $\mathcal{P}$ is the privacy contribution.

Problem Statement & Contributions

We aim to develop a lightweight incentive mechanism that rewards clients who contribute to increasing prediction accuracy based on important features and preserving privacy. The reward function is given by:

$$\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$$

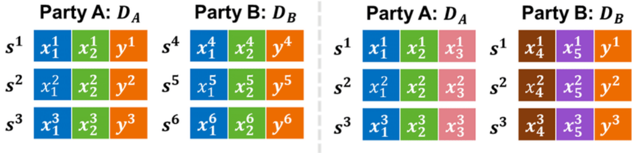
where $\mathcal{T}_i$ is the reward for client i, $\mathcal{I}$ is the performance contribution and $\mathcal{P}$ is the privacy contribution.

Key Contributions:

- Develop a novel incentive mechanism (IM) for VFL that rewards clients for improving prediction accuracy with key feature contributions while upholding privacy.
- Pinpoint features that markedly boost prediction accuracy.
- Ensure the IM's scalability, facilitating VFL efficiency despite tight resource limitations.

Background: Horizontal & Vertical Federated Learning (FL)

FL facilitates training AI models across multiple parties with local data, eliminating the need for data exchange.



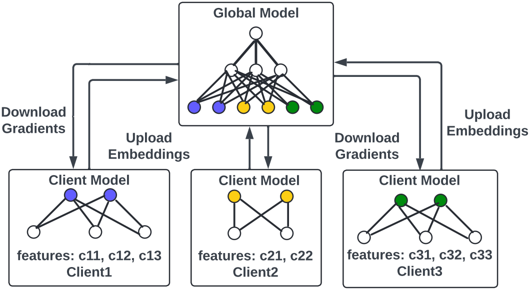
(a) Horizontal FL (b) Vertical FL

FL Types:

- **Horizontal FL (HFL)**: Parties hold data samples from the same sample space but different feature space.
- **Vertical FL (VFL)**: Parties hold data samples from the same feature space but different sample space.

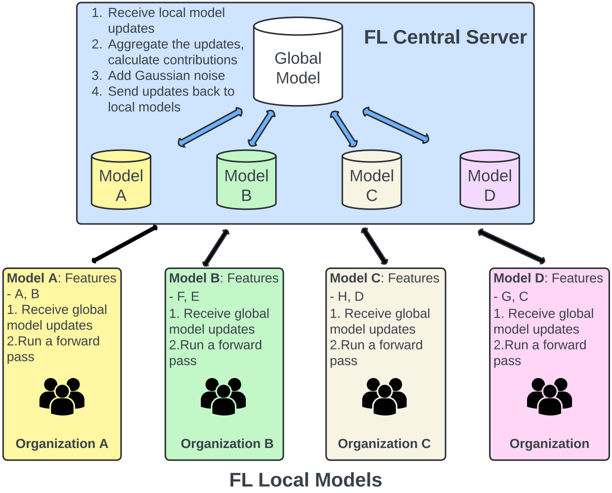
Source: Jiang et al., "Comprehensive analysis of privacy leakage in vertical federated learning during prediction." Proceedings on Privacy Enhancing Technologies (2022).

System Model



- The VFL system includes several clients and a single central server.
- Each client holds a unique subset of features, while the server has labels.
- All clients operate under a semi-honest assumption.
- The server is presumed to be entirely honest.
- Clients typically represent organizations such as medical or educational institutions.

Proposed Framework



Privacy-Preserving Mechanism: Differential Privacy

Overview:

- Optimize Differential Privacy (DP) to preserve a required level of privacy while meeting acceptable prediction accuracy of the FL model.
- Guarantee that the analysis output remains largely unaffected by the presence/absence of a single data entry.
- Tuning key DP parameters, including ε (noise level) and sensitivity.

Proposed Approach:

- The server adds Gaussian noise to the global model update at each iteration.
- The server adjusts noise level based on the privacy preference of clients.

Importance-based Feature Selection

Objectives:

- Reduce overfitting by removing irrelevant or redundant features.
- Improve model interpretability by focusing on influential features.

Importance-based Feature Selection

Objectives:

- Reduce overfitting by removing irrelevant or redundant features.
- Improve model interpretability by focusing on influential features.

Feature Selection Techniques in ML:

- Filter methods: Select features independently.
- Wrapper methods: Use predictive model performance.
- Embedded methods: Feature selection during model training.

Importance-based Feature Selection

Objectives:

- Reduce overfitting by removing irrelevant or redundant features.
- Improve model interpretability by focusing on influential features.

Feature Selection Techniques in ML:

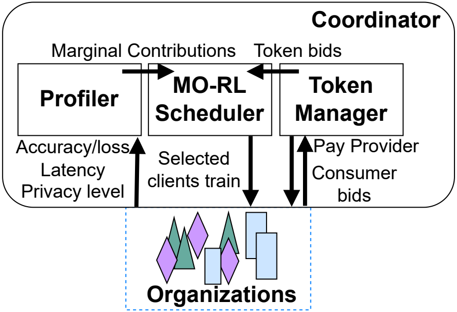
- Filter methods: Select features independently.
- Wrapper methods: Use predictive model performance.
- Embedded methods: Feature selection during model training.

Challenge: Clients do not have access to labels.

Proposed Approach:

- Clients perform a PCA on its features.
- They then pick the features that contribute most to the principle components to participate in the federation.

Proposed Incentive Mechanism



- We adopt a token-based incentive mechanism in our approach.
- Profiler module calculates contributions of each client.
- Token manager handles distribution of tokens.
- Clients are then selected based on their performance contributions.

Proposed Incentive Mechanism (Cont.)

Objective: $$\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$$

$$ClientCost = Unit - Cost \times Memory \times CPU - Utilization$$

Proposed Incentive Mechanism (Cont.)

Objective: $\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$

$$ClientCost = Unit - Cost \times Memory \times CPU - Utilization$$

Reward calculation: for each client $i \in [N]$, and round $r \in [R]$:

$$C_s \leftarrow sort(\mathcal{I}(c_i, \mathcal{D}), \mathcal{P}(c_i, l))$$

$$\beta = N_r \times \frac{(N_r + 1)}{2}$$

$$\tau_i = \tau_i + C_s \times \frac{\tau_{ar}}{\beta} * I_{util}$$

$$\tau_{ar} = \tau_{ar} - \tau_i$$

$$\tau_i = \tau_i + \frac{\tau_{ar}}{N_r}$$

C_s: Rank of clients; I_{util}: Utility improvement of the model accuracy; β: Normalization term; τ_i: Tokens with client i; τ_{ar}: Remaining tokens; N_r: Number of participants; $\mathcal{I}$, $\mathcal{P}$: Performance, privacy contribution.

Proposed Incentive Mechanism (Cont.)

Objective: $\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$

$ClientCost = Unit - Cost \times Memory \times CPU - Utilization$

Reward calculation: for each client $i \in [N]$, and round $r \in [R]$:

$$C_s \leftarrow sort(\mathcal{I}(c_i, \mathcal{D}), \mathcal{P}(c_i, I)) \qquad // \text{ sort by client contribution}$$

$$\beta = N_r \times \frac{(N_r + 1)}{2}$$

$$\tau_i = \tau_i + C_s \times \frac{\tau_{ar}}{\beta} * I_{util}$$

$$\tau_{ar} = \tau_{ar} - \tau_i$$

$$\tau_i = \tau_i + \frac{\tau_{ar}}{N_r}$$

C_s: Rank of clients; I_{util}: Utility improvement of the model accuracy; β: Normalization term; τ_i: Tokens with client i; τ_{ar}: Remaining tokens; N_r: Number of participants; $\mathcal{I}$, $\mathcal{P}$: Performance, privacy contribution.

Proposed Incentive Mechanism (Cont.)

Objective: $\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$

$$ClientCost = Unit - Cost \times Memory \times CPU - Utilization$$

Reward calculation: for each client $i \in [N]$, and round $r \in [R]$:

$$C_s \leftarrow sort(\mathcal{I}(c_i, \mathcal{D}), \mathcal{P}(c_i, l)) \qquad \text{// sort by client contribution}$$

$$\beta = N_r \times \frac{(N_r + 1)}{2} \qquad \text{// token distribution normalization}$$

$$\tau_i = \tau_i + C_s \times \frac{\tau_{ar}}{\beta} * I_{util}$$

$$\tau_{ar} = \tau_{ar} - \tau_i$$

$$\tau_i = \tau_i + \frac{\tau_{ar}}{N_r}$$

C_s: Rank of clients; I_{util}: Utility improvement of the model accuracy;
β: Normalization term; τ_i: Tokens with client i; τ_{ar}: Remaining tokens;
N_r: Number of participants; $\mathcal{I}$, $\mathcal{P}$: Performance, privacy contribution.

Proposed Incentive Mechanism (Cont.)

Objective: $\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$

$ClientCost = Unit - Cost \times Memory \times CPU - Utilization$

Reward calculation: for each client $i \in [N]$, and round $r \in [R]$:

$$C_s \leftarrow sort(\mathcal{I}(c_i, \mathcal{D}), \mathcal{P}(c_i, l)) \qquad // \text{ sort by client contribution}$$

$$\beta = N_r \times \frac{(N_r + 1)}{2} \qquad // \text{ token distribution normalization}$$

$$\tau_i = \tau_i + C_s \times \frac{\tau_{ar}}{\beta} * I_{util} \qquad // \text{ reward distribution}$$

$$\tau_{ar} = \tau_{ar} - \tau_i$$

$$\tau_i = \tau_i + \frac{\tau_{ar}}{N_r}$$

C_s: Rank of clients; I_{util}: Utility improvement of the model accuracy; β: Normalization term; τ_i: Tokens with client i; τ_{ar}: Remaining tokens; N_r: Number of participants; $\mathcal{I}$, $\mathcal{P}$: Performance, privacy contribution.

Proposed Incentive Mechanism (Cont.)

Objective: $\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$

$ClientCost = Unit - Cost \times Memory \times CPU - Utilization$

Reward calculation: for each client $i \in [N]$, and round $r \in [R]$:

$C_s \leftarrow sort(\mathcal{I}(c_i, \mathcal{D}), \mathcal{P}(c_i, l))$ // sort by client contribution

$\beta = N_r \times \frac{(N_r + 1)}{2}$ // token distribution normalization

$\tau_i = \tau_i + C_s \times \frac{\tau_{ar}}{\beta} * I_{util}$ // reward distribution

$\tau_{ar} = \tau_{ar} - \tau_i$ // token allocation

$\tau_i = \tau_i + \frac{\tau_{ar}}{N_r}$

C_s: Rank of clients; I_{util}: Utility improvement of the model accuracy; β: Normalization term; τ_i: Tokens with client i; τ_{ar}: Remaining tokens; N_r: Number of participants; $\mathcal{I}$, $\mathcal{P}$: Performance, privacy contribution.

Proposed Incentive Mechanism (Cont.)

Objective: $\mathcal{T}_i = w_1 \cdot \mathcal{I} + w_2 \cdot \mathcal{P}$

$ClientCost = Unit - Cost \times Memory \times CPU - Utilization$

Reward calculation: for each client $i \in [N]$, and round $r \in [R]$:

$C_s \leftarrow sort(\mathcal{I}(c_i, \mathcal{D}), \mathcal{P}(c_i, I))$ // sort by client contribution

$\beta = N_r \times \frac{(N_r + 1)}{2}$ // token distribution normalization

$\tau_i = \tau_i + C_s \times \frac{\tau_{ar}}{\beta} * I_{util}$ // reward distribution

$\tau_{ar} = \tau_{ar} - \tau_i$ // token allocation

$\tau_i = \tau_i + \frac{\tau_{ar}}{N_r}$ // redistribute remaining tokens

C_s: Rank of clients; I_{util}: Utility improvement of the model accuracy;
β: Normalization term; τ_i: Tokens with client i; τ_{ar}: Remaining tokens;
N_r: Number of participants; $\mathcal{I}$, $\mathcal{P}$: Performance, privacy contribution.

Experimental Setup: Datasets, Comparing Schemes, & Network Structure

- **Datasets**:
  - ADULT income prediction [1]
  - AVAZU click fraud prediction [2]
- **SOTA Comparing Schemes**:
  - TEA for VFL (Lu et al., 2022)
  - FedSDG-FS: A feature selection-based VFL (Li et al., 2023).
  - A vanilla VFL model (Cebellos et al., 2020)
  - IM for VFL using attention aggregation (Yan et al., 2021).
  - feature selection using homomorphic encryption (Jiang et al., 2022).
- **Network Structure**: A VFL model with two clients and a server

[1] https://www.cs.toronto.edu/

[2] https://www.kaggle.com/

Experimental Setup: Hyperparameters for Neural Networks and Differential Privacy

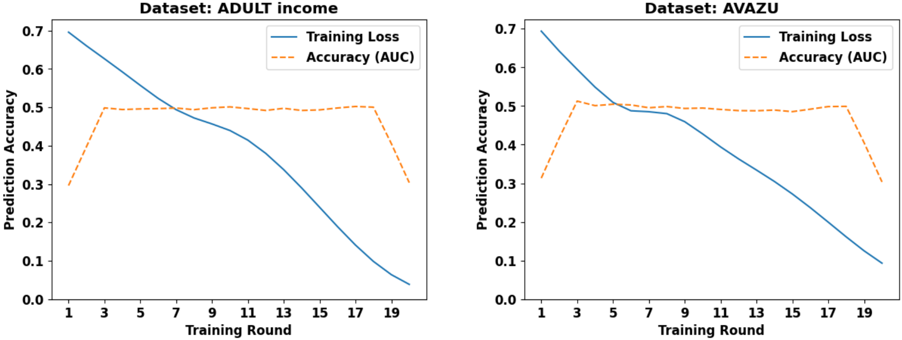
Neural Networks (NNs) are constructed with

- hidden layer size at each client: 128
- hidden layer size at the server: 64
- output dimension: 2
- learning rate: 0.01

DP is parameterized with

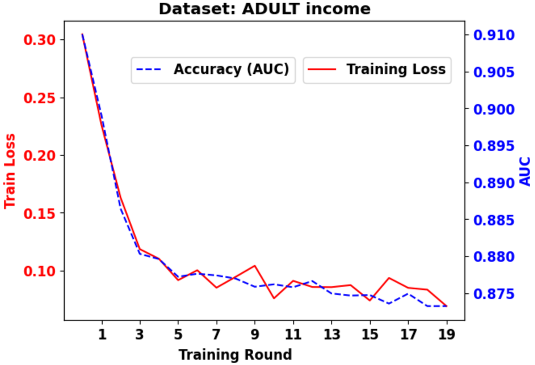
- ε: 0.8
- δ: 1E-6
- sensitivity: 1

Preliminary Results: Impact of PCA Methods on Client's Data:



- When subjected to Differential Privacy (DP), both datasets exhibit identical trends.
- Throughout the training rounds, the training loss consistently declines, while the Area Under the Curve (AUC) metric remains stable.

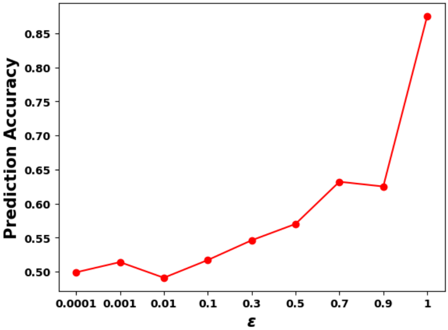
Preliminary Results: Accuracy without Differential Privacy



- Training loss shows similar decreasing trends with or without DP.
- When running without DP, the average prediction accuracy is about 87.5%.

Preliminary Results: Impact of DP

Impact of the parameter ε on model accuracy for ADULT dataset:



- There is a steep increase in prediction accuracy for ε values close to 1.
- Prediction accuracy steadily decreases with decrease in ε.

Key Findings & Future Work

Key Findings:

- PCA and DP do not work well together.
- Adding small amounts of noise significantly reduces model accuracy on our datasets.
- We achieve good accuracies on both our datasets without DP in the vanilla VFL setting.

Key Findings & Future Work

Key Findings:

- PCA and DP do not work well together.
- Adding small amounts of noise significantly reduces model accuracy on our datasets.
- We achieve good accuracies on both our datasets without DP in the vanilla VFL setting.

Future Work:

- Future improvements may involve implementing a more light-weight DP approaches to enhance both model accuracy and training speed.
- Furthermore, employing private collaborative feature selection could contribute to enhancing model performance.

Any Questions?

Thank you!

Contact **Sindhuja Madabushi** at
msindhuja@vt.edu

COLLEGE OF ENGINEERING
COMPUTER SCIENCE
VIRGINIA TECH